

Generative Pretraining

CSE 240D Spring 2026
Ethan Jenkins





Context

NLP before GPT:

- Task-specific models
- Needs labeled data

Problem:

- Poor transfer
- Expensive supervision

QA Task



QA Model

NER Task



NER Model

Sentiment Task



Sentiment Model

Why doesn't learning one NLP task help with another?



So we ask...

**Can one model solve
many tasks?**





Pre-Train and Fine-Tune

Learn language then adapt to tasks





Pre-Training

Predict next word

- Trained on large unlabeled text
- Learns structure, meaning, context
- Self-supervised

"The cat sat on the ____" → "mat"



Pre-Training

Objective: Maximize Likelihood of text

Predict next token given previous tokens

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

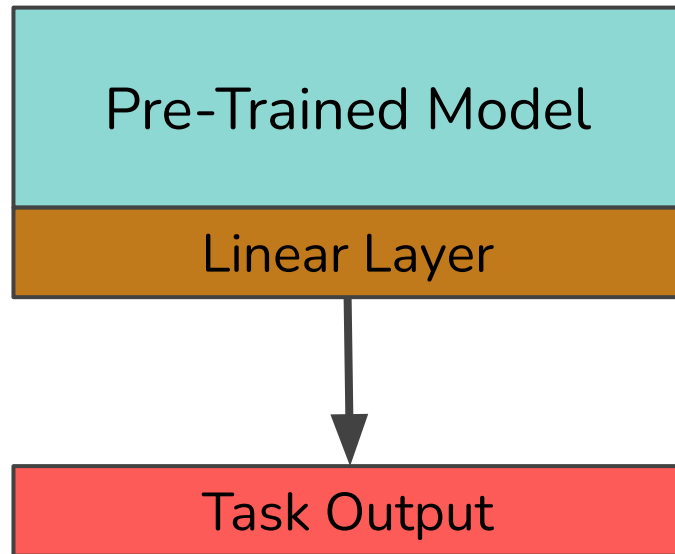
Is predicting the next word enough to learn meaning?



Fine-Tuning

Adapt to a task

- Small labeled dataset
- Add task-specific output layer
- Update all parameters



Starting from pre-trained weights makes learning new tasks much easier



Fine-Tuning

Combine task objective and LM objective

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \dots, x^m)$$

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

What happens if we remove the language modeling objective?



Input Formatting

Modify input for specific tasks by encoding task in the input

Entailment:

[premise] + [hypothesis]

QA:

[context] + [question] + [answer]

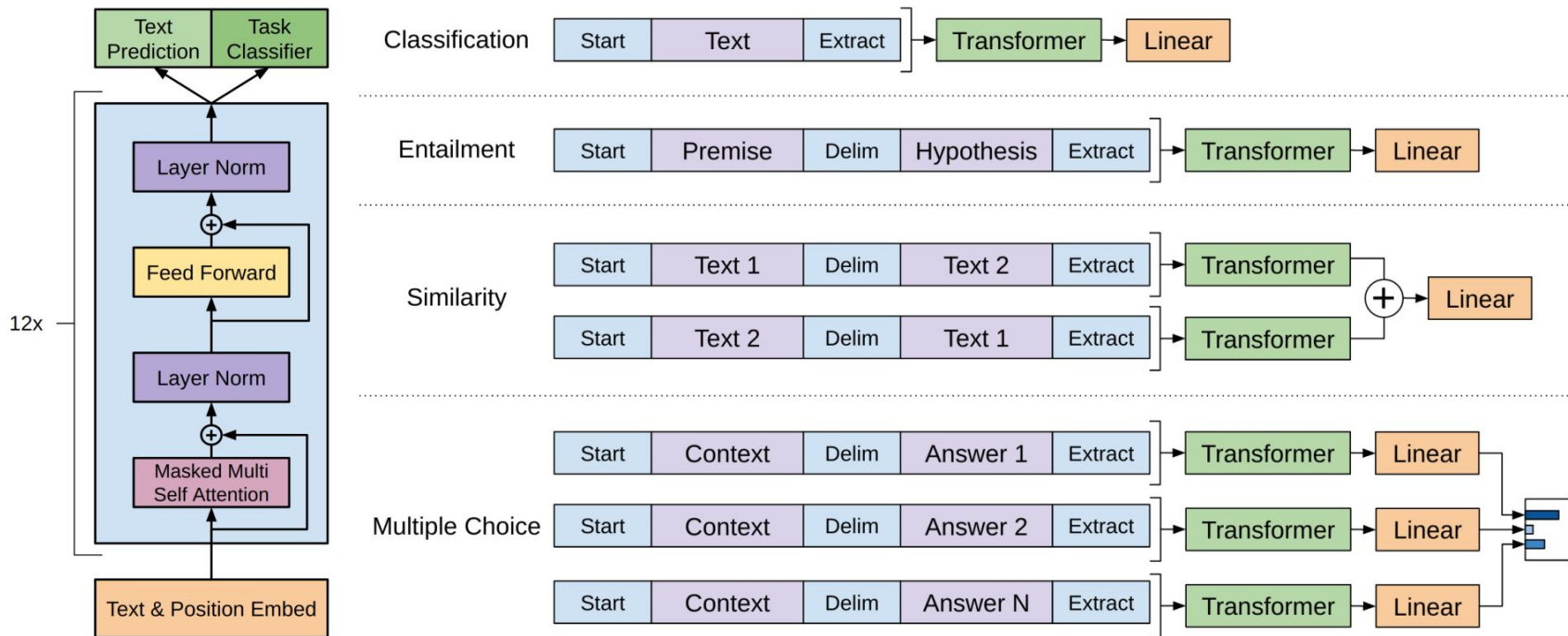
Similarity:

[text1] + [text2]

Same base model, no architecture changes

Why encode the task in the input instead of the model?

Framework Overview



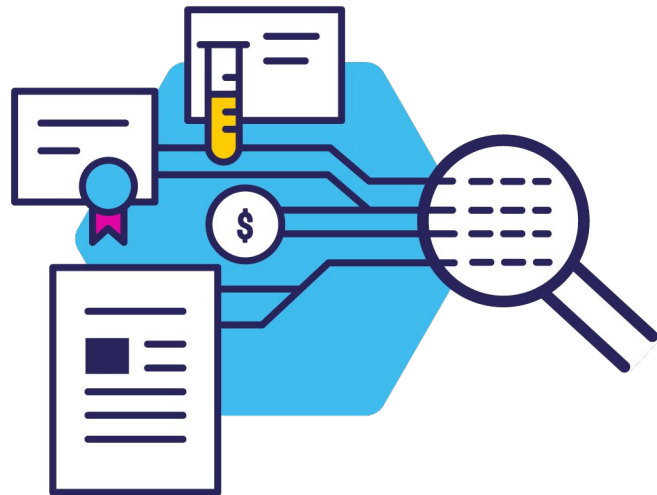
Where does this get us?



Why Pre-training Helps

Better starting point

- Learns general language first
- Reuses knowledge across tasks
- Needs less labeled data
- Easier optimization





Limitations

Not fully general yet

- Requires fine-tuning per task
- Separate output layers
- Input formatting is manual
- Not true zero-shot





Impact

Shift to general-purpose models

- One model $\rightarrow$ many tasks
- Transfer learning becomes standard
- Foundation of modern LLMs



Discussion

- Is language modeling the right objective for all NLP tasks?
- Could we eliminate fine-tuning entirely?
- Is pre-training helping more through better representations or easier optimization?
- What breaks when the data distribution changes?